Congestion Games with Polytopal Strategy Spaces

Hau Chan **Albert Xin Jiang**
Department of Computer Science
Trinity University, San Antonio, TX 78212, USA
{hchan,xjiang}@trinity.edu

Abstract

Congestion games are a well-studied class of games that has been used to model real-world systems such as Internet routing. In many congestion games, each player's number of strategies can be exponential in the natural description of the game. Most existing algorithms for game theoretic computation, from computing expected utilities and best responses to finding Nash equilibrium and other solution concepts, all involve enumeration of pure strategies. As a result, such algorithms would take exponential time on these congestion games. In this work, we study congestion games in which each player's strategy space can be described compactly using a set of linear constraints. For instance, network congestion games naturally fall into this subclass as each player's strategy can be described by a set of flow constraints. We show that we can represent any mixed strategy compactly using marginals which specify the probability of using each resource. As a consequence, the expected utilities and the best responses can be computed in polynomial time. We reduce the problem of computing a best/worst symmetric approximate mixed-strategy Nash equilibrium in symmetric congestion games to a constraint optimization problem on a graph formed by the resources and the strategy constraints. As a result, we present a fully polynomial time approximation scheme (FPTAS) for this problem when the graph has bounded treewidth.

1 Introduction

There has been increasing interest in using game theory to model real-world multiagent systems, with the goal of predicting the behavior of the agents in these settings. This will often require the computation of game-theoretic solution concepts given a description of the model. Nash equilibrium is the most well-known solution concept, and the computational problem of finding Nash equilibria has received extensive study in recent years. While perhaps the most famous theoretical results [Chen *et al.*, 2009; Daskalakis and Papadimitriou, 2005] established the (PPAD-) hardness of the problem given a game in normal form, this does not necessarily mean efficient computation is impossible in practice. In fact, most large games of practical interest are highly structured, and this structure can potentially be exploited for efficient computation.

One of the more well-known class of structured games is *congestion games*, first introduced in [Rosenthal, 1973]. Congestion games model the situation in which there is a set of resources and each agent selects a subset of resources. The cost of using a resource depends on the total number of agents using it. As such, the agent's goal is to select a subset of resources that minimizes the total cost. An important subclass is *network congestion games*, which have been used to model agents' decisions when dealing with (potential) traffic congestion on roads, as well as when sending data through a computer network. More concretely, consider a road network (represented by a graph) and each agent wants to travel from a source to a destination. Since the congestion/delay of each road depends on the number of people driving on them, the goal of each agent is to select a route, from its source to its destination, that minimizes the total delay which is the sum of the delay the agent experience on each road of the route.

A key challenge. Notice that in network congestion games each player's set of pure strategies (i.e. paths) can be exponential in the size of the graph. This is not unique to network congestion games: since a pure strategy is a subset of resources, in many congestion games of interest, the set of pure strategy is exponential in the number of resources. This presents a significant obstacle to efficient computation, as most existing algorithmic approaches for game-theoretic computation assume that the game has *polynomial type* [Daskalakis *et al.*, 2006; Papadimitriou and Roughgarden, 2008], i.e., a polynomial number of players and strategies, because they rely on being able to enumerate the set of pure strategies for each player. As a result, such algorithms would take exponential time on these congestion games.

In this paper, we study congestion games in which each player's strategy space is defined by a *polytope*, specified as set of linear inequalities. This captures a wide range of congestion games; for instance, in the case of network congestion games, we can capture the set of pure strategies of each player using a set of flow constraints [Daskalakis *et al.*, 2006].

In particular, our main contributions are: (i) we show that we can represent players' mixed strategies compactly using marginals. As a result, we can compute the expected utilities

and best responses in polynomial time using the marginals as opposed to the exponential time and space required when using the mixed strategies directly. (ii) We reduce the problem of computing a *best (or worst)* social welfare symmetric approximate Nash equilibrium in symmetric congestion games to a constraint optimization problem on a graph formed by the resources and strategy constraints. (iii) We provide a fully polynomial approximation scheme (FPTAS) when the graph has bounded treewidth. As a corollary, our algorithm can be used to solve network congestion games efficiently when the network has bounded degree and bounded treewidth.

1.1 Related Work

As mentioned earlier, congestion games are first introduced and studied in [Rosenthal, 1973]. It has been shown that every congestion game has a pure-strategy Nash equilibrium (PSNE) via a reduction to its potential game. Despite of this, finding a PSNE is PLS-complete even in network congestion games [Fabrikant *et al.*, 2004]. In the case of symmetric network congestion games, Fabrikant *et al* provided a polynomial time algorithm to find a PSNE. Independent of our work, a recent work [Pia *et al.*, 2015] also studied totally unimodular symmetric congestion games and showed that there is a polynomial time algorithm to compute a PSNE. Furthermore, they also show that it is PLS-complete to find a PSNE in the asymmetric case. In contrast, we focus on finding symmetric approximate mixed-strategy Nash equilibria, which required us to develop novel techniques on compactly representing mixed strategies that is different from the existing literature on congestion games. Furthermore, our algorithm is able to compute extremal equilibria, such as those with the best/worst social welfare, which provide more information about the entire set of equilibria. The only other result for optimal equilibria in congestion games is Ieong *et al*s' algorithm for optimal PSNE in *singleton congestion games*, in which each pure strategy consists of one resource and thus there are only a polynomial number of pure strategies per player [Ieong *et al.*, 2005]. Our algorithm is applicable to a much wider range of congestion games, including those with exponential number of pure strategies.

2 Preliminaries

A *congestion game* is specified by the tuple $(N, R, \{S_i\}_{i\in N}, \{c_r\}_{r\in R}, \{u_i\}_{i\in N})$ where

- $N = \{1, ..., n\}$ is the set of players.
- $R = \{1, ..., m\}$ is the set of resources.
- Each pure strategy $s_i \in S_i$ of player i corresponds to a subset of resources, represented by a $|R|$-dimensional $0-1$ vector. Let s_{ir} be the corresponding component of a resource $r \in R$.
- Let $S = S_1 \times ... \times S_n$ and $s \in S$ be a joint pure strategy of the players.
- $\# : R \times S \to \mathbb{N}$ is a function that counts the numbers of times the resources are used given joint pure-strategies. In other words, $\#(r, s) = \sum_{i\in N} s_{ir}$.
- $c_r : \mathbb{N} \to \mathbb{R}$ is a cost function for resource $r \in R$.
- Given a joint pure-strategy $s = (s_i, s_{-i}) \in S$, the utility of player i is defined to be $u_i(s_i, s_{-i}) = -\sum_{r\in R:s_{ir}=1} c_r(\#(r, s)) = -\sum_{r\in R} s_{ir}c_r(\#(r, s))$.

A *congestion game with polytopal strategy space* (CG^+) is a congestion game $(N, R, \{S_i\}_{i\in N}, \{c_r\}_{r\in R}, \{u_i\}_{i\in N})$ in which S_i is represented as a polytopal strategy space: $S_i = P_i \cap \{0, 1\}^m$ where $P_i = \{x \in \mathbb{R}^m \mid D_i x \leq f_i\}$, $D_i \in \mathbb{Z}^{l_i\times m}$, and $f_i \in \mathbb{Z}^{l_i}$. In other words, P_i is a rational polytope defined by l_i linear constraints. Since S_i should be nonempty for the game to be well-defined, we require that $P_i \cap \{0, 1\}^m \neq \emptyset$ for each $i \in N$.

Before we move on, we make the following assumption.

Assumption 1. *For each player i, the constraint matrix D_i is totally unimodular.* [1]

As a result of this assumption, all extreme points of the polytope P_i are integral (and therefore pure strategies of i). In other words, $P_i = conv(S_i)$.

Network Congestion Games. For instance, network congestion games are an important class of CC^+ that has the totally unimdoular property. A network congestion game of n players is defined on a directed graph $G = (V, E)$ with vertex set V and edge set E. Each player i's goal is to select a path from its source $s_i \in V$ to its destination $d_i \in V$ that minimizes the sum of delay on each edge, which depends on the number of players selects the edge. The set of resources is the set E, and the strategy set of each player i is the possible number of paths from s_i to d_i in G. For each player i, its constraint matrix D_i is the $|V|$ by $|E|$ incidence matrix of G and D_i is totally unimodular [Pia *et al.*, 2015].

Notation and Definition. The set of mixed strategies of player i is defined to be the set of probability distributions over i's pure-strategy set S_i. We let $X_i = \Delta(S_i)$ to denote such set and let $X = X_i \times ... \times X_n$ be the joint mixed-strategy set. Moreover, for any $x_i \in X_i$, $x_i(s_i)$ denote the probability of playing pure-strategy $s_i \in S_i$ and $\sum_{s_i\in S_i} x_i(s_i) = 1$. Let $x(s) = \prod_{i\in N} x_i(s_i)$. Given a joint mixed strategy $x \in X$, the expected utility of player i is defined to be $E_{s\sim x}[u_i(s_i, s_{-i})] = \sum_{s\in S} x(s)u_i(s_i, s_{-i})$. With a slight abuse of notation, we let $u_i(x_i, x_{-i}) = E_{s\sim x}[u_i(s_i, s_{-i})]$. For the rest of the paper, we assume the utility values $u_i(s)$ are all between zero and one.

For ease of reference, we provide the following standard definition of Nash Equilibrium in terms of best-response.

Definition 1. *Given a CG^+ and a joint mixed strategy $x_{-i} \in X_{-i}$ of other players except i, the mixed strategy x_i is an ϵ-best-response of player i if and only if $u_i(x_i, x_{-i}) \geq u_i(\bar{x}_i, x_{-i}) - \epsilon$ for each $\bar{x}_i \in \Delta(S_i)$ and some $1 \geq \epsilon \geq 0$.*

Definition 2. *Given a CG^+, a joint mixed strategy $x \in X$ is an ϵ-mixed-strategy Nash equilibrium (ϵ-MSNE) if and only if for each player $i \in N$, playing x_i is an ϵ-best-response to x_{-i}. We have an exact MSNE when $\epsilon = 0$.*

It is not hard to see that computing best responses is a key computational step in finding an MSNE, and computing expected utilities is an important part of finding best responses.

[1] A matrix is totally unimodular if each of its square sub-matrices has determinant of 1, 0, and -1. [Hoffman and Kruskal, 1956]

However, there are two key issues in using the traditional approach to compute expected utilities. The first is that we need to specify the probability of playing each pure strategy, and, in order to this, we need to know the set S_i for each player i. While finding a solution that is in the polytope takes polynomial time, finding and enumerating all of the feasible solutions would take exponential time. Second, computing expected utilities of a given joint mixed strategy by directly summing over the joint pure strategies could take time exponential in the number of players.

In the next section we describe a way to *compactly* represent each mixed strategy (without enumerating all of the pure strategies), which then allows us to compute the expected utilities more efficiently.

3 Representation of Mixed-Strategies

It turns out that we can use *marginal* vectors to compactly represent mixed strategies. In particular, given a mixed strategy $x_i \in X_i$ of player i, we define the marginal vector $p_i^{x_i} = (p_{ir})_{r \in R}$ to be an m-dimensional vector such that $p_{ir}^{x_i} = \sum_{s_i \in S_i} s_{ir} x_i(s_i)$. In other words, $p_{ir}^{x_i}$ specifies the probability that i plays resource r given the mixed strategy x_i. When the context is clear, we will drop the superscript x_i. As a result, we can show that we can compute the expected utility of each player using the marginal mixed-strategies.

Claim 1. *Given a joint mixed-strategy $x_{-i} \in X_{-i}$ of other players except i and a pure-strategy $s_i \in S_i$, the expected utility of i of playing s_i can be written as* $u_i(s_i, x_{-i}) = -\sum_{r \in R} \sum_{s_{-i} \in S_{-i}} s_{ir} c_r \left(\sum_{j \in N \setminus \{i\}} s_{jr} + 1 \right) x_{-i}(s_{-i})$.

Claim 2. *Given a joint mixed strategy $x_{-i} \in X_{-i}$ of other players except i and a pure strategy $s_i \in S_i$, let $p_j = (p_{jr})_{r \in R}$ be the marginal vectors of the player $j \in N \setminus \{i\}$ such that $p_{jr} = \sum_{s_j \in S_j} s_{jr} x_j(s_j)$. Let $\Omega_{jr} = \{0, 1\}$ to be the outcome of whether player j will play a pure strategy that contains the resource r. The expected utility of i of playing s_i can be written as*

$$u_i(s_i, x_{-i}) = -\sum_{r \in R} \sum_{\omega_r \in \Omega_r} s_{ir} c_r \left(\sum_{j \in N \setminus \{i\}} \omega_{r_j} + 1 \right) \prod_{j \in N \setminus \{i\}} p_{jr}^{\omega_{r_j}} (1 - p_{jr})^{1 - \omega_{r_j}},$$

where $\Omega_r = \prod_{j \in N \setminus \{i\}} \Omega_{jr}$ is the Cartesian product of the sets $(\Omega_{jr})_{j \in N \setminus \{i\}}$.

Lemma 1. *Given a joint mixed-strategy $x \in X$, the expected utility of player i can be computed via the following*

$$u_i(x_i, x_{-i}) = -\sum_{r \in R} p_{ir} \sum_{\omega_r \in \Omega_r} c_r \left(\sum_{j \in N \setminus \{i\}} \omega_{r_j} + 1 \right) \prod_{j \in N \setminus \{i\}} p_{jr}^{\omega_{r_j}} (1 - p_{jr})^{1 - \omega_{r_j}}. \quad (1)$$

The above shows that we can compute the expected utilities of the players by using marginal vectors. Directly computing the above sum requires $m|\Omega_r| = m2^{n-1}$ summands. Nevertheless, expected utilities can be computed in polynomial time, using a dynamic programming approach analogous to [Papadimitriou and Roughgarden, 2008; Jiang *et al.*, 2011].

From Marginal Probabilities to Mixed-Strategies Let us now consider the issue of constructing a mixed strategy given a marginal vector. First of all, since we have assumed that the extreme points of the polytope P_i are integer points, and thus $P_i = conv(S_i)$, this becomes the problem of describing a point in a polytope by a convex combination of extreme points of the polytope. By Caratheodory's theorem, given $\pi_i \in \mathbb{R}^{m_i}$ there exists a mixed strategy of support size at most $m_i + 1$ that matches the marginals. There has been existing work that provides efficient constructions for different types of polytopes, including the Birkhoff-von Neumann theorem and its generalizations [Budish *et al.*, 2013]. The most general result in [Grötschel *et al.*, 1981] reduces the problem to the task of optimizing an arbitrary linear objective over P_i. This can be solved via linear programming in time polynomial in m and l_i (the number of constraints in P_i).

Thus, instead of specifying the probabilities of the mixed-strategies, which can be exponential, now we only need to specify the marginal probabilities of playing the resources.

3.1 Linear Programming Formulation of Best Responses

As part of our algorithmic approach for finding MSNE, we would like to formulate the best response condition as a set of constraints on the player's marginal vectors. One formulation is that the expected utility of playing the best response should be greater or equal to the expected utilities of playing all other pure strategies. However, even though we can compute expected utilities efficiently, stating the above will require an exponential number of inequality constraints. Instead, we will reformulate the best-response condition of each player using linear programing.

Let $p_i = (p_{ir})_{r \in R}$ be a marginal vector of player i. Let $\nabla_i(p_{-i}) = (U_r(p_{-i}))_{r \in R}$ be the (resource-utility) vector that specifies the contributions to i's utility from each resource given that the other players play the strategy p_{-i}. Then the expected utility of i (the expression in Lemma 1) can be written as $p_i^T \nabla_i(p_{-i})$. Therefore, given the marginal vectors of other players, the best-response for player i is the solution to the following linear program.

$$\begin{array}{ll} \text{maximize} & p_i^T \nabla_i(p_{-i}) \\ \text{subject to} & D_i p_i \leq f_i \\ & 1 \geq p_{ir} \geq 0 \text{ for } r \in R \end{array}$$

We can formulate the dual linear program as the following.

$$\begin{array}{ll} \text{minimize} & f_i^T \lambda_i \\ \text{subject to} & D_i^T \lambda_i \geq \nabla_i(p_{-i}) \\ & \lambda_{ij} \geq 0 \text{ for } j = 1, 2, ..., l_i \end{array}$$

Since the primal LP is feasible, by LP duality the primal and dual LPs obtain the same optimal objective value. Therefore, p_i is Player i's best response when there exists a feasible dual vector λ_i such that $p_i^T \nabla_i(p_{-i}) = f_i^T \lambda_i$. Thus, a feasible marginal vector $\bar{p}$ is an MSNE if for each player i, there exists $\lambda_i \geq 0$ such that $D_i^T \lambda_i \geq \nabla_i(\bar{p}_{-i})$ and $\bar{p}_i^T \nabla_i(\bar{p}_{-i}) = f_i^T \lambda_i$.

Therefore, to check whether a particular marginal vector is a best-response, we can check to see whether the expected utility is equal to the optimal solution of the dual program.

This will involve only a polynomial number of constraints and variables, in contrast to the exponential number of constraints in the direct formulation. Our goal now is to find a set of marginal probabilities that will correspond to an MSNE.

Notice that the players' linear programming formulation of best responses do not tell us how to find an equilibrium, thus we provide an algorithm to find one next.

4 Optimal Approximate Symmetric MSNE

Most algorithmic results for congestion games focus on finding a PSNE, including polynomial algorithms for symmetric network congestion games [Fabrikant *et al.*, 2004] and symmetric CG^+ [Pia *et al.*, 2015]. Nevertheless, finding one equilibrium would have limited usefulness since there can be an exponential number of equilibria in a game. We instead focus on finding extremal equilibria, which tell us more information about the entire set of equilibria. In particular, we consider the problem of computing a *best/worst* social welfare symmetric ϵ-MSNE efficiently for symmetric CG^+. Our results and algorithm can be extended to k-symmetric CG^+ with constant number of player types k.

Definition 3. *Consider a CG^+ with $(N, R, \{S_i\}_{i\in N}, \{c_r\}_{r\in R}, \{u_i\}_{i\in N})$. The game is symmetric if and only if $D_i = D_j = D$, $f_i = f_j = f$ for all $i, j \in N$* [2].

Below, we state a known useful result [Nash, 1951].

Proposition 1. *For every symmetric CG^+, there is a symmetric MSNE in which every player plays the same strategies.*

Since the same mixed strategies can be represented using the same marginal vectors, the above implies the existence of symmetric MSNE in terms of marginal vectors.

While there can be multiple symmetric MSNE in a symmetric CG^+, we are interested in those with maximal (or minial) social welfare, where social welfare is defined to be the sum of all players' expected utilities. Since in a symmetric MSNE of a symmetric game, all players have the same expected utility, this is equivalent to finding symmetric MSNE with maximal/minimal utility for one player.

Next, we show the existence of a symmetric ϵ-MSNE in which the marginal vector lies in a discretized grid in P_i. This allows us to search for ϵ-MSNE in this discretized space.

Lemma 2. *For any $\delta > 0$, there is a (symmetric) marginal vector $q = (q_r)_{r\in R}$ in which $q_r \in \{0, \frac{1}{K}, \frac{2}{K}, \ldots, 1\}$ and $K = O(\frac{\log m}{\delta^2})$, such that $u(q, q_{-i}) = q^T\nabla(q_{-i}) \geq q'^T\nabla(q_{-i}) - O(\delta nm)$ for $q' \in [0,1]^m$ that satisfies $Dq' \leq f$.*

Proof. Recall that there is at least one $s \in S^*$ such that $Ds \leq f$. It follows from Proposition 1, there is a symmetric MSNE $x^* \in X$ in any symmetric CG^+. Let $p = (p_r)_{r\in R}$ be the marginal vector of the MSNE. From the approximated version of the Caratheodory's theorem (Theorem 3 of [Barman, 2015]), for every $\delta > 0$ and for every $p \in conv(S)$, there is a multi-set $Q \subset S^*$ of size $O(\frac{\log m}{\delta^2})$ and $q = \frac{1}{|Q|}\sum_{\bar{q}\in Q}\bar{q}$ and $q \in conv(S^*)$ such that $||p - q||_\infty \leq \delta$. Thus, there is a marginal mixed-strategies q such that $|p_r - q_r| \leq \delta$ for all $r \in R$. Next, we provide the following useful claims.

[2] This implies $S_i = S_j = S^*$, $P_i = P_j = P^*$ for all $i, j \in N$.

Claim 3. *Given $r \in R$, $|\nabla(q_{-i})_r - \nabla(p_{-i})_r| \leq O(\delta n)$.*

Proof. (Sketch) From Lemma 1, the difference of the expected utility from a resource r can be written as $|\nabla(q_{-i})_r - \nabla(p_{-i})_r| \leq \sum_{\omega_r\in\Omega_r} \left|\prod_{j=1}^{n-1} q_r^{\omega_{r_j}}(1-q_r)^{1-\omega_{r_j}} - \prod_{j=1}^{n-1} q_r^{\omega_{r_j}}(1-q_r)^{1-\omega_{r_j}}\right|$, because the utility/cost functions are between 0 and 1. It can be shown that the total variation distance of the above distribution is $O(\delta n)$ [Daskalakis *et al.*, 2009]. □

Claim 4. *For any marginal mixed-strategy p, $|p^T\nabla(p_{-i}) - p^T\nabla(q_{-i})| \leq O(\delta nm)$.*

Claim 5. *The marginal symmetric MSNE p is an $O(\delta nm)$-best-response to the marginal mixed-strategy q when all other players play according to q.*

Proof. By the definition of the symmetric MSNE, for any $p' \in conv(S)$, $p^T\nabla(p_{-i}) \geq p'^T\nabla(p_{-i})$. Moreover, $p^T\nabla(q_{-i}) + O(\delta nm) \geq p^T\nabla(p_{-i}) \geq p'^T\nabla(p_{-i}) \geq p'^T\nabla(q_{-i}) - O(\delta nm)$, where the first inequality is by Claim 4, the second inequality is by the definition of MSNE, and the last inequality is by Claim 4. It follows that $p^T\nabla(q_{-i}) \geq p'^T\nabla(q_{-i}) - O(\delta nm)$. Our result follows since p is no less than the expected utility of other marginal mixed-strategies minus $O(\delta nm)$. □

To finish the proof, notice that, by our construction/existence of q, for all $r \in R$, $|p_r - q_r| \leq \delta$. Thus,

$$q^T\nabla(q_{-i}) + O(\delta m) \geq p^T\nabla(q_{-i}) \geq p'^T\nabla(q_{-i}) - O(\delta nm)$$

and the last inequality is by Claim 5. Therefore, q is an $O(\delta nm)$-best-response to all others using the marginal mixed-strategies q. If every player plays according to q, then we have an $O(\delta nm)$-MSNE. □

4.1 Conditions for Symmetric ϵ-MSNE

From the proof of Lemma 2, we know the possible (discretized) values of q. However, the crucial part is checking whether each marginal mixed-strategy (of the discretized space) can be used to form an ϵ-MSNE.

While we could use the standard ϵ-MSNE conditions for a given marginal mixed-strategy, it would take exponential time. Instead of using the conditions, we will use the solution of the dual linear program (as discussed in Section 3.1) to check for equilibrium. We will derive such condition below.

For tractability, we also discretize the resource-utility space and project the resource-utility to some discretization of the space. Let $V = \{v_i, i = 0, 1, 2, ..., \lceil\frac{4m}{\epsilon}\rceil \mid v_i = \frac{i\epsilon}{4m}\}$ to be the discretization of the utility space for some epsilon $1 \geq \epsilon > 0$.

Claim 6. *Given a symmetric marginal mixed-strategy q, and let $proj(u) = v$ be the projection of some value u to a value $v \in V$ such that $u \in (v - \frac{\epsilon}{4m}, v + \frac{\epsilon}{4m})$. We have $\sum_{r\in R} q_r proj(\nabla(q_{-i})_r) \geq \sum_{r\in R} q'_r proj(\nabla(q_{-i})_r) - \frac{\epsilon}{2} - O(\delta nm)$, for any $q' \in [0,1]^m$ that satisfies $Dq' \leq f$.*

Proof. To verify the above claim, notice that, for each $r \in R$, $|\nabla(q_{-i})_r - proj(\nabla(q_{-i})_r)| \leq \frac{\epsilon}{4m}$, and, for any marginal mixed-strategy $\bar{q}$, $\left|\sum_{r\in R} \bar{q}_r \nabla(q_{-i})_r - \sum_{r\in R} \bar{q}_r proj(\nabla(q_{-i})_r)\right| \leq \frac{\epsilon}{4}$. Thus, our bound follows from the above inequality. □

Using the above claim, we provide the following lemma.

Lemma 3. *Given a symmetric marginal vector q, and let $proj(u) = v$ be the projection of some value u to a value $v \in V$ such that $u \in (v - \frac{\epsilon}{4m}, v + \frac{\epsilon}{4m})$. Let q^* be a best response to q. We have $\sum_{r\in R} q^*_r proj(\nabla(q_{-i})_r) = f^T\lambda^* - \frac{\epsilon}{2} - O(\delta nm)$, where $f^T\lambda^*$ is the optimal solution of the dual problem that corresponds to the primal program given the discretized utility values under $proj(\cdot)$* [3].

The above lemma allows us to verify whether a marginal vector can be used to form an ϵ-MSNE. It is not clear how we can find the λ^* as claimed in the above lemma. Below, we provide a lemma to specify the potential values of λ^*.

Lemma 4. *The value of $\lambda^* = (\lambda^*_i)_{i=1,2,...,l}$ is such that $\lambda^*_i = c_i v_i$ for some integer constant $0 \leq c_i \leq m$ and $v \in V$.*

Proof. Since the solution for the primal program (i.e., existence of an ϵ-MSNE) exists, there is a solution for its dual. Since the transpose D^T is totally unimodular by Assumption 1, each λ^*_i is some linear combination of the value of $proj(\nabla(q_{-i}))$. Moreover, V is a uniform discretization and $\lambda^*_i \geq 0$. Thus, λ^*_i is some combination of values in V. □

4.2 An Algorithm to Find Symmetric ϵ-MSNE

In this subsection, our goal is to develop an efficient algorithm to find marginal vectors that satisfy the ϵ-MSNE conditions as discussed in the previous subsection.

Table 1 lists the variables and their uniform discretization that we used in the following discussion. To get an ϵ-MSNE, we set $\delta = \frac{\epsilon}{O(nm)}$. Notice that the marginal q_r of each $r \in R$ has finitely many values (from Lemma 2) in Q. Since the marginals have finitely many values and the resource-utility values are discretized and projected as specified in the previous subsection, the expected utility has finitely many values in T. In addition, because the constraint matrix consists of only 0, 1, and -1, for any constraint c, the possible values of $c \cdot q$ of any $q \in Q^m$ is in ∂D. Finally, each dual variable λ_i is positive and is some linear combination of values in Q. As a result, $\lambda_c \in \Lambda$ for some constraint c.

A natural question is how can we use these discretized values to compute ϵ-MSNE efficiently? Before answering this question, we define and introduce the following notation.

Definition 4. *Given a symmetric CG^+ with $(N, R, \{S\}_{i\in N}, \{c_r\}_{r\in R}, \{u\}_{i\in N})$. Let C be the set of indices of constraints in P^* and $c \in C$ corresponds to row c of D. The game's* constraint graph *is a directed graph $G = (R \cup C, E)$ where $E = \{(c, r) \in C \times R \mid D_{cr} \neq 0\}$. The game's* induced graph *is an undirected graph $IG = (R, E)$ where $E = \{(r_1, r_2) \in R^2 \mid \exists j \in C, D_{j,r_1} \neq 0, D_{j,r_2} \neq 0\}$.*

We now state our main result.

[3] To get the primal/dual programs, take the programs in Section 3.1 and replace $\nabla_i(p_{-i})$ by $proj(\nabla_i(p_{-i}))$.

Table 1: Variables (Var.) and Discretization

Var.	Discretization
q_r	$Q = \{q_i, i = 0, ..., O(\frac{\log m}{\delta^2}) \mid q_i = \frac{i}{\lvert Q\rvert}\}$
λ_c	$\Lambda = \{\lambda_i, i = 0, ..., \frac{4m^2}{\epsilon} \mid \lambda_i = \frac{i\epsilon}{4m}\}$
t	$T = \{t_i, i = 0, ..., \frac{4m^2 O(\frac{\log m}{\delta^2})}{\epsilon} \mid t_i = \frac{i\epsilon}{4mO(\frac{\log m}{\delta^2})}\}$
∂d	$\partial D = \{\pm q_i, i = 0, ..., mO(\frac{\log m}{\delta^2}) \mid q_i = \frac{i}{O(\frac{\log m}{\delta^2})}\}$

Theorem 1. *Given a symmetric CG^+ whose constraint graph's in-degree is bounded by a constant d and whose induced graph has treewidth bounded by a constant w. There is a fully polynomial time approximate scheme (FPTAS) to compute a best/worst social welfare symmetric ϵ-MSNE in $poly(n, m, \frac{1}{\epsilon})$ time.*

As a corollary, we can state the sufficient condition for FPTAS as a single graphical property of the game. Given a directed graph (R, E), its *primal graph* (also known as moral graph) is the undirected graph $(R, E \cup \{(r, w) | \exists t \in R, (r, t) \in E, (w, t) \in E\})$, i.e., adding edges between each pair of in-neighbors of each node.

Corollary 1. *Given a symmetric CG^+ such that the treewidth of the primal graph of the constraint graph is bounded by a constant W. There is a fully polynomial time approximate scheme (FPTAS) for compute a best/worst social welfare symmetric ϵ-MSNE in $poly(n, m, \frac{1}{\epsilon})$ time.*

For simplicity of exposition, we describe our algorithm for the case when the in-degrees of the constraint graph are bounded by d and when its IG is a tree. The algorithm can be generalized to IG with a bounded tree-width W.

Without loss of generality, we orient the tree in "level-order" so that the first level consists of the root, the second level consists of its children and so on. For convenience, we relabel the nodes/resources so that the nodes are ordered such that root is n, and its children, $Ch(n)$, are (from right to left) $n - |Ch(n)|$, ..., $n - 1$, and subsequently for their respective children in the respective level. We let $D(r) = \{j \in C \mid D_{jr} \neq 0\}$ to be the set of constraints such that their column corresponding to resource r is nonzero. Since $\nabla(q_{-i})_r$ for $r \in R$ only depends on q_r, we write $\nabla(q_r)_r$ instead.

Our message passing algorithm will be run on the induced graph. Each resource will pass the appropriate messages to its neighbors as specified below, first in an upstream pass from the leaves to the root, then in a downstream pass from root to the leaves. Since each resource, say a $r \in R$, can be involved in more than one constraints (each of which corresponds to a dual variable), we need to keep track of possible values of $(\lambda_j)_{j\in D(r)}$ and the possible (sum of) values of $D_{:,r}q_r$. Indeed, since D is totally unimodular and we have a discretization on q_r, we have discretization on the possible sums of the constraints. It is clear that we have a finite discretization for the possible value of λ as well. As a result, this reduces to a constraint optimization problem on discrete values.

To begin, we first describe the message from a leave $l \in$

R of the tree to its parent. Notice that along each message passing, we will keep track of the best possible value of $\lambda^* = (\lambda_j^*)_{j\in C}$. We begin by discussing the upstream pass.

Upstream Pass. For each leave $l \in R$, l computes a table $f_l : Q \times T \times \partial D^{|D(l)|} \times \Lambda^{|D(l)|} \to \{0,1\}$ such that

$$f_l(q_l, t, (\partial d_j)_{j\in D(l)}, (\lambda_j)_{j\in D(l)}) = 1[t = q_l proj(\nabla(q_l)_l)]$$
$$\prod_{j\in D(l)} 1[\partial d_j = D_{jl} q_l] 1[D^T_{lD(l)} \cdot (\lambda_j)_{j\in D(l)} \geq proj(\nabla(q_l)_l)]$$

specifies the feasible ϵ-MSNE region. Moreover, the size of this table is $poly(n, m, \frac{1}{\epsilon})$ (for bounded in-degrees). Let $k = Pa(l)$ denote the parent of l and let $D(l,k) = D(l) \cap D(k)$. The message of $l \to k$ is

$$M_{l\to k} = \{(q_l, t, (\partial d_j)_{j\in D(l,k)}, (\lambda_j)_{j\in D(l,k)}) \mid \exists f_l(q_l, t, (\partial d_j)_{j\in D(l)}, (\lambda_j)_{j\in D(l)}) = 1 \text{ and } \forall j \in D(l) \setminus D(l,k)\ \partial d_j \leq f_j\},$$

is the set of feasible tuples of size at most $poly(n, m, \frac{1}{\epsilon})$. In addition, for $j \in D(l) \setminus D(l,k)$, $\lambda_j^* \in \arg\min_{(\cdot)\in M_{i\to k} : f(\cdot,\lambda_j)=1} f_j \lambda_j$. For each non-leaf node $r \in R$, r computes a table $f_r : Q \times T \times \partial D^{|D(r)|} \times \Lambda^{|D(r)|} \to \{0,1\}$ given the messages from its children. Let $Ch(r)$ be the children of r and and we order the children by their index such that $r_1 < r_2 < ... < r_{|Ch(r)|}$. Notice that r will have $|Ch(r)|$ messages and the goal is to combine these messages efficiently and construct f_r.

Below, we describe an efficient way of combining the messages. For $k = r_1$, we construct

$$f_{r,k} = \{(q_r, t, (\partial d_j)_{j\in D(r,k)}, (\lambda_j)_{j\in D(r)}) \mid \forall (q_k, t_k, (\partial d'_j)_{j\in D(r,k)}, (\lambda_j)_{j\in D(r,k)}) \in M_{k\to r}$$
$$1[t = t_k + q_r proj(\nabla(q_r)_r)] \prod_{j\in D(r,k)} 1[\partial d_j = \partial d'_j + D_{jr} q_r]$$
$$1[D^T_{rD(r)} \cdot (\lambda_j)_{j\in D(r)} \geq proj(\nabla(q_r)_r)]\},$$

and for $k = r_2, ..., r_{|Ch(r)|}$, we construct the following successively, in that order,

$$f_{r,k} = \{(q_r, t, (\partial d_j)_{j\in \bigcup_{i=r_1}^{k-1} D(r,i) \cup D(r,k)}, (\lambda_j)_{j\in D(r)}) \mid \forall (q_r, t', (\partial d_j)_{j\in \bigcup_{i=r_1}^{k-1} D(r,i)}, (\lambda_j)_{j\in D(r)}) \in f_{r,k-1}$$
$$\forall (q_k, t_k, (\partial d'_j)_{j\in D(r,k)}, (\lambda_j)_{j\in D(r,k)}) \in M_{k\to r}$$
$$1[t = t' + t_k] \prod_{j\in D(r,k)} 1[\partial d_j = \partial d'_j + D_{j,r} q_r]$$
$$1[D^T_{rD(r)} \cdot (\lambda_j)_{j\in D(r)} \geq proj(\nabla(q_r)_r)]\}.$$

Finally,

$$f_r(q_r, t, (\partial d_j)_{j\in D(r)}, (\lambda_j)_{j\in D(r)}) = 1[(q_r, t, (\partial d_j)_{j\in \bigcup_{k\in Ch(r)} D(r,k)}, (\lambda_j)_{j\in D(r)} \in f_{r,r_{|Ch(r)|}}] \prod_{j\in D(r)\setminus \bigcup_{k\in Ch(r)} D(r,k)} 1[\partial d_j = \partial d'_j + D_{j,r} q_r].$$

Similarly, the message r will send to its parent k is

$$M_{r\to k} = \{(q_r, t, (\partial d_j)_{j\in D(r,k)}, (\lambda_j)_{j\in D(r,k)}) \mid \exists f_r(q_r, t, (\partial d_j)_{j\in D(r)}, (\lambda_j)_{j\in D(r,k)}) = 1 \text{ and } \forall j \in D(r) \setminus D(r,k)\ \partial d_j \leq f_j\},$$

is the set of feasible tuples. In addition, for $j \in D(r) \setminus D(r,k)$, $\lambda_j^* \in \arg\min_{(\cdot)\in M_{r\to k} : f(\cdot,\lambda_j)=1} f_j \lambda_j$.

Note that computing the $f_{r,k}$ requires us to deal with at most two tables of $poly(n, m, \frac{1}{\epsilon})$ size. Thus, computing the final table f_r doesn't blow up the computation exponentially.

Downstream Pass. After root n has received the messages from its children, it is going to compute a partial ϵ-MSNE and pass it to its children. Given f_n, we construct

$$R_n = \{(q_n, t, (\partial d_j)_{j\in D(n)}, (\lambda_j)_{j\in D(n)}) \mid f_n(q_n, t, (\partial d_j)_{j\in D(n)}, (\lambda_j)_{j\in D(n)}) \prod_{j\in D(n)} 1[\partial d_j \leq f_j]\}.$$

For $j \in D(n)$, let $\lambda_j^* \in \arg\min_{(\cdot,\lambda_j)\in R_n} f_j \lambda_j$. Next, find [4]

$$(q_n, t, (\partial d_j)_{j\in D(n)}, (\lambda_j)_{j\in D(n)}) \in \arg\max_{(\bar{q}_n, \bar{t}, (\overline{\partial d}_j)_{j\in D(n)}, (\bar{\lambda}_j)_{j\in D(n)})\in R(n)} 1[\bar{t} \geq f^T \lambda^* - \epsilon]\bar{t}.$$

Indeed, n will send such a message to each of its children. Given the message from the parent, say i, $k = r_{|Ch(i)|}, ...r_2$, in that reverse order, will compute

$$R_{i,k}(q_i, t_{k+1}, (\partial d_j)_{j\in D(i)}, (\lambda_j)_{j\in D(i)}) = \{(q_k, t_k, (\partial d'_j)_{j\in D(k,i)}, (\lambda'_j)_{j\in D(k,i)}) \in M_{k\to i} \text{ and } (q_i, t_{k+1} - t_k, (\partial d'_j)_{j\in \bigcup_{i'=r_1}^{k-1} D(i,i')}, (\lambda'_j)_{j\in D(i)}) \in f_{i,k-1} : \prod_{j\in \bigcup_{i'=r_1}^{k} D(i,i')} 1[\partial d'_j = \partial d_j - D_{ji} q_i] \prod_{j\in D(i')} 1[\lambda'_j = \lambda_j]\},$$

where k will select (an arbitrary tuple of) message in $R_{i,k}$ and send it to its children and the $k-1$ node. In particular, k will send the $M_{k\to i}$ tuple to its children and send the $f_{i,k-1}$ to $k-1$ so that $k-1$ will use that tuple to compute $R_{i,k-1}$. When $k = r_1$, we have

$$R_{i,k}(q_i, t_{k+1}, (\partial d_j)_{j\in D(i)}, (\lambda_j)_{j\in D(i)}) = \{(q_k, t_k, (\partial d'_j)_{j\in D(k,i)}, (\lambda'_j)_{j\in D(k,i)}) \in M_{k\to i} \text{ and } 1[t_{k+1} - t_k = q_i proj(\nabla(q_i)_i)] \prod_{j\in D(k,i)} 1[\partial d'_j = \partial d_j - D_{ji} q_i] 1[\lambda'_j = \lambda_j]\}.$$

Standard techniques that generalize algorithms for tree to those of bounded treewidth yield Theorem 1.

Computing ϵ-MSNE in Symmetric network congestion games. As discussed in Section 2, the constraint matrix of the players in network congestion games is totally unimdoular. It turns out that the induced graph of the constraint matrix is the line graph [5] of G. Applying a bounded-treewidth result in [Atserias, 2008; Bienstock, 1990; Călinescu *et al.*, 2003], if the graph of a congestion game has constant treewidth and degree, the treewidth of its line graph is bounded.

Corollary 2. *There is an FPTAS to compute a best/worst symmetric ϵ-MSNE for congestion games where their graphs have constant treewidth and constant degree.*

[4] For the worst ϵ-MSNE, replace $\arg\max$ by $\arg\min$.

[5] Given a graph $G = (V, E)$, a line graph of G is a graph on E where two edges in E are connected if they are incident in G.

References

[Atserias, 2008] A. Atserias. On digraph coloring problems and treewidth duality. *European Journal of Combinatorics*, 29(4):796 – 820, 2008.

[Barman, 2015] S. Barman. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of caratheodory's theorem. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 361–369, 2015.

[Bienstock, 1990] D. Bienstock. On embedding graphs in trees. *Journal of Combinatorial Theory, Series B*, 49(1):103 – 136, 1990.

[Budish *et al.*, 2013] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom. Designing random allocation mechanisms: Theory and applications. *The American Economic Review*, 103(2):585–623, 2013.

[Călinescu *et al.*, 2003] G. Călinescu, C. G. Fernandes, and B. Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. *Journal of Algorithms*, 48(2):333 – 359, 2003.

[Chen *et al.*, 2009] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *J. ACM*, 56(3):14:1–14:57, May 2009.

[Daskalakis and Papadimitriou, 2005] C. Daskalakis and C. H. Papadimitriou. Three-player games are hard. *ECCC, TR05-139*, 2005.

[Daskalakis *et al.*, 2006] C. Daskalakis, A. Fabrikant, and C. H. Papadimitriou. The game world is flat: The complexity of nash equilibria in succinct games. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006*, pages 513–524, 2006.

[Daskalakis *et al.*, 2009] C. Daskalakis, G. Schoenebeck, G. Valiant, and P. Valiant. On the complexity of nash equilibria of action-graph games. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 710–719, 2009.

[Fabrikant *et al.*, 2004] A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 604–612, 2004.

[Grötschel *et al.*, 1981] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[Hoffman and Kruskal, 1956] A. Hoffman and J. Kruskal. Integral boundary points of convex polyhedra, in Linear Inequalities and Related Systems(H. Kuhn and A. Tucker, Eds.). *Annals of Maths. Study*, 38:223–246, 1956.

[Ieong *et al.*, 2005] S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *AAAI*, volume 5, pages 489–494, 2005.

[Jiang *et al.*, 2011] A. X. Jiang, K. Leyton-Brown, and N. Bhat. Action-graph games. *Games and Economic Behavior*, 71(1):141–173, January 2011.

[Nash, 1951] J. F. Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, 1951.

[Papadimitriou and Roughgarden, 2008] C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14, July 2008.

[Pia *et al.*, 2015] A. D. Pia, M. Ferris, and C. Michini. Totally unimodular congestion games. *CoRR*, abs/1511.02784, 2015.

[Rosenthal, 1973] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.